Data Analysis of The Lancaster Project

Becky Lytle

May 3, 2016

1 Introduction

My data set, "The Lancaster Project," was made up of data collected in a class taught during the fall of 2014 by history professor Andrew Friedman. This class was called "Walter Benjamin on Lancaster Avenue: A History of American Modernity." Throughout the semester, students in this class were expected to add items to this data set that reflected the archival techniques of Walter Benjamin, a German-Jewish philosopher and critical theorist.

First, I had combed through the various attributes in this data in order to decide which would stay, which could be ignored, and what attributes I could possibly add to the data set. The attributes that I removed from this data set were coverage, collectionId, collectionResource, collectionURL, extendedResourcesExhibitPages, featured, filesResource, filesURL, itemTypeId, itemTypeName, itemTypeResource, itemTypeURL, ownerResource, ownerURL, public, and URL. I removed these either due to all the data having the same value at an attribute or because of the attribute being irrelevant in terms of data analysis (such as URL). The attributes that I used (from the original data set) were date, description, rarity, sense of the modern, source, title, date added, filesCount, ID, itemTypeName, date modified, ownerID, and tags. I turned tags, which was a column that had words and phrases associated with each point in the data set, into several columns of numerical data. Each individual tag became its own column, and if a point had that tag associated with it then it had a 1 in that column, and a 0 otherwise. I also turned "itemTypeName" into four numerical columns: person, place, thing, and event. Additionally, I added new attributes based on the existing ones. These included weekdayAdded, a number representing the day of the week that an item was added; time, the time that an item was added; and syllabusWeek, a number representing the week of the semester that an item was added. Also, the Rarity scale was initially on a scale from 1 to 10, but I normalized it so that the scale was from 1 to 5 instead, in order to be in line with the Sense of the Modern scale, which was from 1 to 5.

Next, I had to decide what to do in terms of the missing values in my data. When cleaning my data set, I chose to replace missing values with items' nearest neighbors' values. I immediately ruled out filling items' missing values with the mean for that column, because this inherently wouldn't make sense for

categorical data because you can't take the average of categorical data. Then, I had to choose between filling missing values with nearest neighbors' data or filling with a random value from that column, and the former option sounded more appealing for our data because it would potentially make the filled items' values more accurate.

There were a few problems with this, considering a couple items had missing titles and descriptions; however, the amount of items with missing values in these columns was low enough to make filling missing values with nearest neighbors' data worth it, even though filling an items' title or description with another items' title or description clearly would not be accurate. There are a few implications with filling missing values with nearest neighbors' values. This means that items that had a lot of missing values, when filled with nearest neighbors' data, would be almost identical to this nearest neighbor. This somewhat makes this nearest neighbors' values more "important" because then they appear more times in the data. The same situation would occur if a certain point happens to be a lot of points' nearest neighbor. I think that the possible downfalls of this method of data cleaning when doing data analysis are overridden by the possibility of accuracy when filling missing values when compared to the accuracy of filling missing values with a random value from that column.

Now, I will explain exactly what I mean when I say that I "filled missing values with the values of the nearest neighbor." The "nearest neighbor" of a certain item in the data set is the one that is "closest" to it in terms of distance. The distance metric I used to find the nearest neighbor of an item and also used throughout my data analysis for The Lancaster Project was euclidean distance, which is defined by the following formula.

$$d(p,q) = \sqrt[2]{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2} \tag{1}$$

If p and q are two different items in the data set, then q_i and p_i represent the values of these two items at the i^{th} column. This distance scale only works with numerical data (as opposed to some other distance metrics that take categorical data into account), so that is why I converted some of my categorical data into numerical data, as described above. This distance metric sometimes downplays the distance between points that only differ in terms of, for example, a tag, because each tags' column either has a 0 or a 1 in every row, and the distance between 0 and 1 is not very large. It can also put more importance on attributes such as Sense of the Modern that are a scale from 1 to 5. However, this fact is actually why I made Rarity a scale from 1 to 5 instead of 1 to 10, because otherwise a difference in Rarity would have meant more than a difference in Sense of the Modern in terms of distance. Therefore, I was somewhat able to work around the limits of euclidean distance.

In order to find the nearest neighbor of a certain item, and therefore find the item with which to replace missing values, I first constructed a KD-Tree with all the items in my data set that did not have missing values. A KD-Tree, otherwise known as a binary space partitioning tree, splits based on the median of a group of items in a data set in one dimension, and alternates dimension to

use for the "split point" for each level of the tree.

So, for example, the root node for a set of n points would be the median of these points based on the 1st columns' attribute. Assuming that n is an odd number (for the sake of this example), there would be $(n-1)/2$ items that have a lower value in the 1st columns' attribute (let's call this set A) and $(n-1)/2$ items that have a higher value in the 1st columns' attribute (let's call this set B). The left child of this initial root node will be the median of set A at the 2nd columns' attribute, and the right child of this initial root node will the median of set B at the 2nd columns' attribute. This pattern continues with the number of the attribute increasing with each level of the tree.

The process of making a KD-Tree stops when a certain node has only one possible left child (or no possible left child) and only one possible right child (or no possible right child). This could potentially happen if a node is a median of three points, so the list of points that are less than this point at the attribute being used to split would only include one point, and the list of points that are greater than this point at the attribute being used to split would also only include one point. Then, this node would have a left and a right child and both of these children would have no children of their own. Also, a quick side note about alternating dimensions to split by: if there are x attributes in a certain data set, then the x^{th} level of the tree will use the x^{th} attribute to split, while the $(x+1)^{th}$ attribute of the tree will use the 1st attribute to split.

After I made a KD-Tree with all my data, I did a query through the tree for each of my items with missing data. I will explain the concept of a query by explaining the various steps involved. For the sake of my explanation of querying through a KD-Tree, let the item that has missing data be called item g and let this item's nearest neighbor be called "best" while the distance to its nearest neighbor is called "bestD." The following steps represent the steps necessary in a query.

1. Set "best" equal to the rootnode of the KD-Tree and set "bestD" equal to the distance to this rootnode.

2. If the rootnode does not have children, proceed to step 4. Compute the distance between the rootnode's left child and item g. If it does not have a left child, move onto step 3. Make sure to only use the attributes that both of these items actually have in order to take the distance; for example, if item g is missing attribute i, then you'd take the distance using all attributes except for attribute i.

    - If this distance is less than "bestD," set "bestD" equal to this distance and set "best" equal to the rootnode's left child. Repeat step 2, letting the new rootnode be the original rootnode's left child.
    - If this distance is not less than "bestD", and the rootnode does have a right child, then move on to step 3.
    - If this distance is not less than "bestD" and the rootnode does not have a right child, proceed to step 4.

3. Compute the distance between the rootnode's right child and item g.

    - If this distance is less than "bestD", set "bestD" equal to this distance and set "best" equal to the rootnode's right child. Repeat step 2, letting the new rootnode be the original rootnode's right child.
    - If this distance is not less than "bestD", move onto step 4.

4. When you are sent to this step, it is because your rootnode does not have children, or if your rootnode's children are not closer to item g than the rootnode. Recall what you last saved as your "best" and what distance you last saved as "bestD." Now, go back up the tree checking each subtree's rootnode that you haven't yet checked by finding the distance between it and item g. If the distance is smaller than "bestD" then follow the above steps for this subtree after setting the subtree's rootnode equal to "best" and its distance to item g as "bestD." Once you've done this for every subtree that you hadn't yet looked at, return your values for "best" and "bestD." "Best" is the nearest neighbor of item g.

Using the above steps, I found each item with missing values' nearest neighbor and used it to fill in these missing values.

Often when doing data analysis, people have questions from the domain expert that guide their exploration. In my case, Professor Friedman simply wanted to find anything in the data, as if the data had its own story to tell us. It seemed as if he was searching for some sort of narrative that explained what was happening with the data. From this, I formed some of my own questions about the data that helped me choose what to explore in terms of data analysis. The following questions are what guided my research:

- Are there any attributes in this data set that are correlated? For example, is there a certain level of Sense of the Modern that correlates with the use of a certain tag? Or, were items with a high Rarity posted in a certain set of syllabus weeks?
- Does the importance of a certain item as designated by the students and professor of this class correlate with an item's PageRank? Does an item's importance in a network correlate to its relative importance in the eyes of the students?

2 Clustering by K-Means

One data analysis technique that I used to analyze this data set was clustering by k-means. There are many different ways to "cluster" data. The goal of clustering is essentially to separate data into k different clusters that show something about the data, whether it be to show what points are similar to one another, what points are outliers, or other types of discoveries. The theory behind k-means

clustering is that k clusters will be found that minimize the sum of the squared pairwise within-cluster distances.

The algorithm for k-means clustering starts by picking k centers, usually represented by k items in the data set. I will later explain what number I used for k and why, but for now, I will just use k to represent any possible number of clusters. In order to find these k centers, I used the k-center clustering algorithm. First, I set p to any point in my data set. Then, I added p to my set of centers C. Until the set C had k centers, I set p equal to the furthest point from any point in C and then added that point to C. In order to find the furthest point from any point in C, I found the average value for every attribute using the points already in C, and then I calculated the distance from that new point to every other point in the data set, allowing me to find the one that was furthest away.

Using these k centers, I was able to proceed with the k-means algorithm. I then began the main step of the algorithm. Let's call this step 2. Once I picked these centers, I assigned each point in the data set to its nearest center. For each cluster, I computed a new center by finding the "centroid" of the points. The centroid of a cluster is a point that takes the average value at every attribute for all of the points in the cluster. I repeated step 2 until the clusters reached a "steady state." A steady state was reached when the centers stopped changing significantly between each repetition of step 2. By "stopped changing significantly," I mean that each center was remaining within a range of 5 percent error for each repetition of step 2.

After writing this algorithm, I had to figure out how many clusters I actually wanted my data to cluster into; in order words, I had to find the correct k value to use. In order to do this, I had to run k-means clustering on values of k from 1 to $\sqrt[2]{n/2}$, with n being the number of items in our data set; in my case, $n = 365$. Using $\sqrt[2]{n/2}$ as the upper limit when finding the value of k to use just happens to be the rule of thumb when clustering a set of data. For every run of k-means with a different value of k, I recorded each clusters' radius and outputted the largest radius out of these. I kept track of the largest radii for each k value by putting them into a list. To find the value of k to use, I found the "elbow" of this list by finding the largest drop between radii (this is called the "elbow method" of finding k). The biggest drops in my list were from $k = 1$ to $k = 2$, and from $k = 2$ to $k = 3$. I therefore decided to use $k = 3$, because out of the options of $k = 1$, 2, and 3, I felt that $k = 3$ would yield the most interesting results.

After I ran k-means clustering on my data set using $k = 3$, I ran an algorithm that would find labels to best represent each cluster based "differential cluster labeling." The idea is that each label would accurately describe what a certain cluster's points had in common. The concept behind this strategy was to find a label for a cluster that would maximize "mutual information" for that cluster. Mutual information is the amount that knowing X reduces uncertainty about Y, if X and Y are two different attributes. The equation for mutual information for two attributes X and Y is shown below.

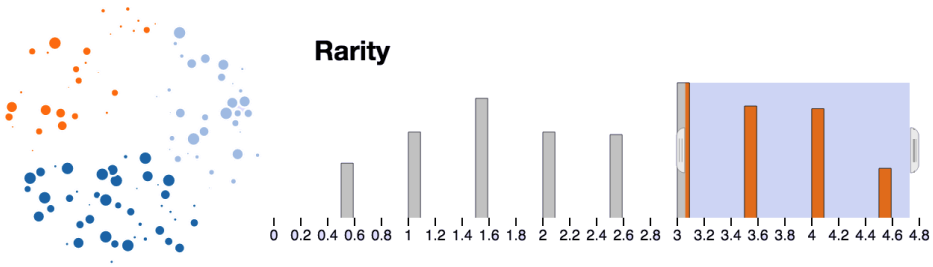
$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2}$$

Using the differential cluster labeling algorithm, each cluster was given a label with an attribute and a value for that attribute. For example, one cluster was given the label of "ownerID: 4."

After I ran my k-means clustering algorithm and the differential clustering labeling algorithm, I created a visualization to go along with the information generated by this data analysis. The following figure shows the three clusters that my data naturally clustered into using the k-means clustering algorithm. The labels on the right of the visualization correlate with the cluster that has the same color as the the label.

The first cluster, itemTypeThing:1, is labeled this because the item type for these (in general) is a thing (rather than a person, place, or event). This represents roughly 1/3 of the data. The second cluster, date:12/3/2014, has a lot of points that were put onto the system on this date. This date is close to the end of the semester, so I am sure a lot of students were trying to get all their work done at the end, and that evidently involved entering in data to this data set. The third cluster, ownerId:4, shows that one of the students in this class really loved this project and posted a lot of items in this data set that were very similar to one another. For a student to be important enough to become the label of a cluster is fairly interesting.

After viewing this visualization, I decided that I would filter the clusters to only include points with a Rarity of 3 or higher. The following visualization represents this filtering.



When filtering the above clustering to include points with a rarity of 3 or higher, there are more points that appear in the cluster defined by data points with an item type of "thing" when compared with the other two clusters. Not only does this cluster have more points shown when filtering out lower rarities, but it also has more large points, showing that the data in this cluster has a higher frequency of rarities on the higher end of the rarities being shown. Overall, this visualization demonstrates the fact that data points with the item type "thing" have a higher chance of being more rare than other data points. After speaking with Professor Friedman about this finding, I learned that Walter Benjamin actually had a theory about "things" (as opposed to people, places, or events) having certain attributes in terms of obscurity and tangibility and that this may play into the fact that students actually found these "things" to be more rare than other types of items.

Some future work that I would want to do in relation to what I found here might be to see if there are any other attributes that relate to other item types, such as person, place, or event. It would also be interesting to see if an item's PageRank (which I'll explain later) correlates to its item type.

3 Network Analysis and PageRank

For my second data analysis technique, I created a network out of my data set, calculated the PageRank for each item in the data, and then created a visualization in order to analyze these results. I had initially done this midway through the semester, but I was using my original nearest neighbors algorithm that turned out to be inaccurate; as I will explain in a moment, nearest neighbors are vital to the creation of my network, so it was important that my nearest neighbor algorithm was fixed. Therefore, before proceeding with the following steps, I fixed my nearest neighbor algorithm to query through a KD-Tree correctly. I also used this fixed nearest neighbor algorithm when cleaning my data, as described in the introduction of this paper.

In order to create the network, I connected each item to 5 of its nearest neighbors. I found these nearest neighbors using the method that I described in the introduction. I also did not include the attribute "ID" when calculating nearest neighbors because I did not feel that it was relevant to an items' similarity to another point, considering each point had a different ID. I saved each items' list of nearest neighbors; let's call these the"out" connections of these items, because these connections are from the item itself to its nearest

neighbors. Then, using this information, I was able to compute what the "in" connections were for each item. For example, if one of item a's nearest neighbors is item b, then one of item b's "in" connections is item a. Basically, the "in" connections were found by observing which "out" connection lists an item was in.

Then, using this information, I was able to calculate each item's PageRank. PageRank is actually an algorithm that Google once used to determine the importance of each web page on the internet, and because the internet is essentially just one large network, I was able to compute PageRank for each item in my network. In order to compute PageRank, I first set each item's PageRank to $1/n$, with n being the number of items in the data set. In the future, when I reference the "original" PageRank of an item, I am referencing these PageRanks.

Then, using this information, I used the following formula to calculate each item's new PageRank.

$$PR(u) = \frac{1-d}{n} + d \sum_{v \in In(u)} \frac{PR(v)}{\mid Out(v) \mid} \tag{3}$$

Let In(v) = set of "in" connections that an item v has, and let Out(v) = set of "out" connections than an item v has. Let PR(v) be the original PageRank of item v, and let PR(u) be the new PageRank of item u. Also, d is the "damping factor" which represents the probability that you keep going through the network (or, in terms of the internet, the probability that you keep clicking on links to get to new pages). Google found that d is usually equal to 0.85 in order for the PageRank equation to work, and they added $\frac{1-d}{n} + d$ as the "teleportation" part of the equation.

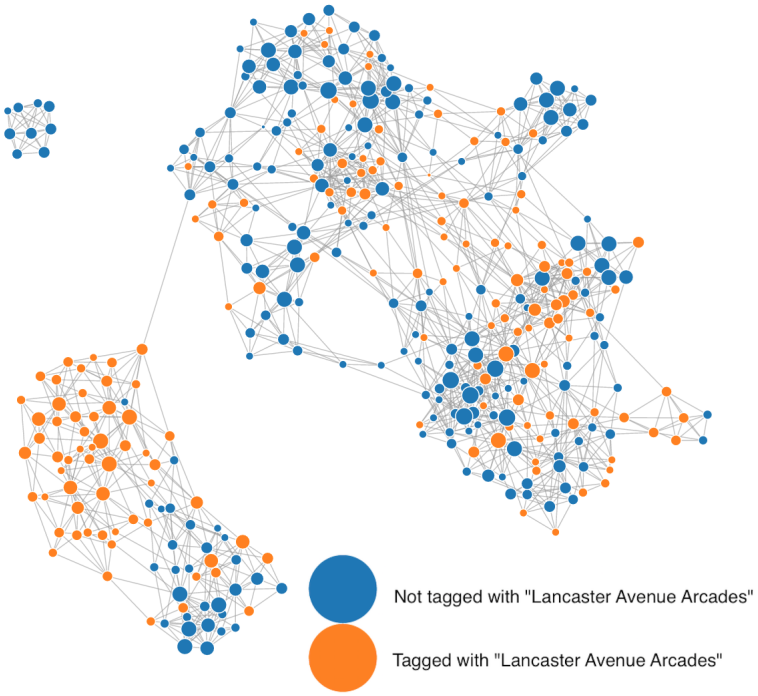
After I calculated the new PageRanks for each item, I saved all of these as the new "original" PageRanks and repeated the above step to calculate each items' new PageRank again. I repeated this process until each items' PageRank stopped significantly changing. By "stopped significantly changing," I mean that the PageRanks started to remain within a 1 percent range of error as the above step continued repeating.

Then, I created a visualization using the information from my network creation and PageRank calculations. I decided to size each item's node in the visualization based on its PageRank; if an item had a higher PageRank, I made its node larger in order to represent its importance. Then, I had to decide how to color each node.

Initially, when I first used this data analysis technique, I just tried coloring the different items in the data by various numerical data, such as Rarity, Sense of the Modern, syllabus week, and others. For example, if I was coloring by Rarity, then items with a Rarity of 1 would have a different color from items with a Rarity of 2, and so on.

However, there were so many numerical columns (considering there is a column for each tag) that I wasn't able to try all of them in order to color my network. When speaking to Professor Friedman at my poster session, I found

out that the tag "Lancaster Avenue Arcades" represented something significant. Apparently, at the end of the semester during the fall of 2014, when Professor Friedman's class about Walter Benjamin was taught, the class had to decide which items from this data set would be shown in an exhibit. They debated which items were most important and which truly represented the spirit of Walter Benjamin; the 166 items that were chosen to be in the exhibit were tagged with "Lancaster Avenue Arcades." Once I had learned this, I was interested as to whether or not the importance of each item (as calculated by PageRank) was related to the importance of each item as decided by Professor Friedman's class. I then decided to color the nodes in my network based on whether or not they were tagged with "Lancaster Avenue Arcades." The following visualization shows this network.



As you can see, the network essentially forms into two general areas: one larger and one smaller. There is one small area of points in the top left corner; however, these points are simply from when the class was getting a tutorial on how to enter items into this data set and are therefore negligible.

The smaller group of points (in the bottom left corner of this network visualization; group A) are mostly comprised of points from the first few weeks of the class during Fall 2014, while the larger group of points (group B) is comprised of points from the middle and end of the semester. It is clear that the composition of these groups is different in terms of which points are tagged with "Lancaster Avenue Arcades" and which aren't.

When looking at group B, it is clear that the orange points (items tagged with "Lancaster Avenue Arcades") are the smaller points with less connections. This means that they are the points with smaller PageRanks, and therefore, less "importance" as determined by PageRank. However, PageRank determines importance by surveying the connections that an item has to and from other items. Therefore, during the middle and end of the semester, the most important points as determined by Professor Friedman's class were the ones with the least connections, which meant that they were the most unique items that did not have many other items like them. During this part of the class, PageRank had an inverse relationship with the class's classification of certain points as "important"; in general, if a point had a lower PageRank, it was more important.

On the other hand, group A was mostly comprised of points that were tagged with "Lancaster Avenue Arcades," and these points had a relatively higher PageRank than the other points in this group. These points are also highly connected to one another. Therefore, at the beginning of the semester, if items had a lot of connections to other points in the data (and thus, had a large PageRank), they were more likely to be chosen as "important" by the class later in the semester. This pattern is essentially opposite to the pattern observed later in the semester.

These patterns are especially interesting because they occur in very specific time periods, rather than randomly throughout the semester. It is possible that these patterns occurred due to what was expected from students at different points in the semester. At the beginning of the class, it is possible that what was deemed a "good" item in the data set was something concrete that was related to the class's introduction to Walter Benjamin. This would explain why all the important points (as chosen by the class) in the beginning of the semester were very similar to one another. Likewise, throughout the middle and the end of the semester, it would make sense for students to be expected to start making more unique data entries drawing from Benjamin's theories, considering people's understanding of a topic naturally deepens as time goes on. These theories might explain why the class ended up choosing certain points as "important" and worth putting in an exhibit later on.

Future work based on these findings might involve seeing if there were any ways to predict whether or not a point would be chosen to have the tag of "Lancaster Avenue Arcades" based on its other attributes, such as Rarity, Sense of the Modern, and tags. If there was a way to predict this, it could potentially help any future students in Professor Friedman's class if he were to teach this seminar again.

4 Conclusion

My first data analysis technique, k-means clustering, not only helped visualize my data set in a new way, but it helped me draw a conclusion about the correlation of two attributes. It was helpful to see what groups my data naturally clustered into when k was set equal to three, considering the labels of these

clusters demonstrated the various attributes that defined these different clusters. Overall, the big takeaway from this data analysis appeared when I filtered Rarity to include items with a Rarity of 3 or higher in my visualization. When doing this, it was apparent that items with an item type of "thing were more likely to have a higher Rarity than items with an item type of "person," "place," or "event." This conclusion shows a clear correlation between two attributes in this data set.

My second data analysis technique, network analysis and PageRank computation, also helped to visualize this data in a new way by showing how items connected to their five nearest neighbors while also helping show how PageRank relates to the tag of "Lancaster Avenue Arcades. Because we know that this tag represented a sort of importance considering items tagged with this were shown in an exhibit after Professor Friedmans class ended, it was interesting to see how this interacted with another kind of importance: importance determined by PageRank. Through this data analysis technique, it was shown that in the beginning of the semester, points that were later deemed important by the class were all very similar to one another, as if they were all inspired by a similar introductory topic; these points had a relatively high PageRank. On the other hand, points later in the semester that were deemed important by the class were very unique and did not have many connections in the network other than to their own five nearest neighbors; these points had a relatively low PageRank. These patterns tell a sort of narrative about how the students and professor of this class determined the importance of certain points and how this importance related to PageRank at different points during the semester.

These conclusions bring up new questions that may guide future work with this data set. As Ive stated previously, building off of my conclusions from my PageRank analysis, I would like to see if it is possible to predict whether or not an item would be tagged with "Lancaster Avenue Arcades" based on the values of its other attributes. Another thing Id like to explore that relates to the network created from my data is if there are any words (in an items description or title) that are associated with a higher PageRank and what these words are. Overall, there are many more routes to take in order to analyze this data set further and I would have liked to continue exploring with The Lancaster Project.

References

[1] Sorelle Friedler. *datavizcourse* (2016). BitBucket repository. http://bitbucket.org/sorelle/datavizcourse/src.

[2] Mike Bostock. *Force-Directed Graph* (2016). http://bl.ocks.org/mbostock/4062045

[3] Andrew Friedman. *Walter Benjamin on Lancaster Avenue: Syllabus* (2014). History Dept., Haverford College, Haverford, PA. Microsoft Word file.